

WebKit 内核源代码分析 (二) ---dlmu2001

tomorrow.cyz@gmail.com

摘要：本系列通过分析 WebKit 的源代码，试图分析 WebKit 的内核设计架构，模块之间的关系，分析的时候以 Qt 的移植为参考，涉及移植的东西不多，主要还是以内核为主。FrameLoader 类负责一个 Frame 的加载，在 Frame 的流程中起到非常重要的重要，同很多组件都有交互，本文将分析 FrameLoader 类的代码。

1. 概述

顾名思义，FrameLoader 是一个 Frame 的 loader，它的作用就是为客户提供一个下载一个 Frame 的一系列接口。这里的客户指的是类的客户，比如 Frame 类，间接客户是上层应用，比如 qwebframe。

从它的定义看，最容易想到的是一个 load 接口，用来将一个 frame load 下来。任何一个页面至少都需要一个 mainframe，因此一个页面的下载一般就是从 load 一个 mainframe 开始。

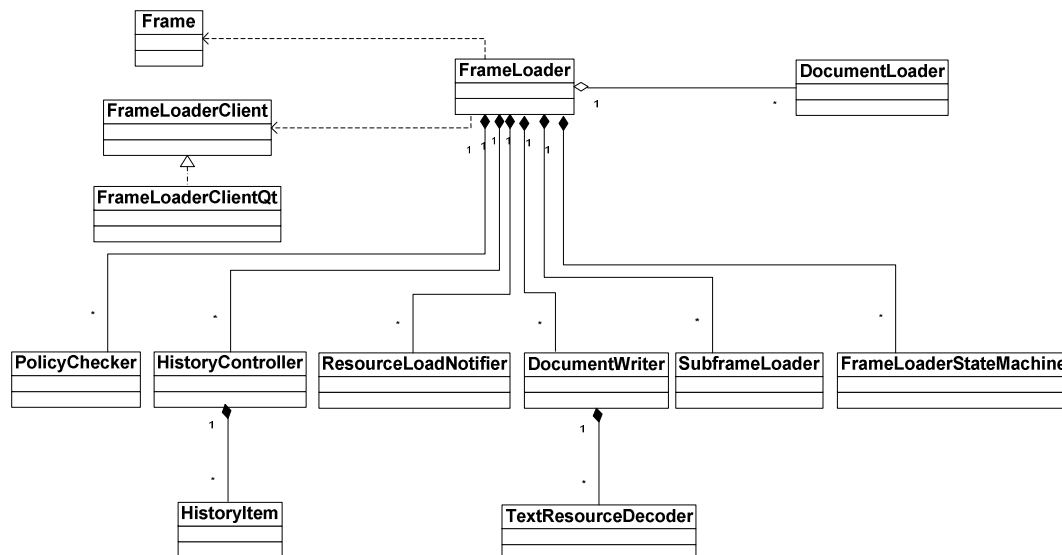
在 load frame 的过程中，通过 FrameLoaderClient 接口将 load 过程的不同阶段告知客户。

FrameLoader 通过 setDocumentLoader 相当于把 load 的工作委托给了 DocumentLoader 类。

FrameLoader 同 DocumentLoader 是 has-a 的关系。一般在 load 的时候创建 DocumentLoader。Frame 调用 DocumentLoader 的

startLoadingMainResource 开始 load frame。

2. 类关系



1) Frame 和 FrameLoader 是 contain-a 的关系，在 Frame 的构造函数中调用 FrameLoader 的构造函数，调用时传入参数 Frame 指针和 FrameLoaderClient 指针。

2) Frame 有可能有子 Frame，所以维护 SubFrameLoader 对象 m_subframeLoader 来管理子 Frame 的 load。Frame 可以对应 xml document，也可对应 html document，等等。跟 Document 相关的子 resource 的 load 不在 FrameLoader 的职责范围内。

3) 包含一个 DocumentWriter 类对象 m_writer，当 Frame 的数据 load finish 的时候，将数据传给 DocumentWriter 类，进行下一步的处理(比如解码)

4) FrameLoader 维护了三个 DocumentLoader 对象，分别对应于不同

的阶段，m_policyDocumentLoader 对应于收到用户 load 调用，进行 policy check 阶段，m_provisionalDocumentLoader 对应于 policy check 通过以后，Frame 数据还没有到来之前，它会负责 startLoadingMainResource 的调用。m_documentLoader 则是 Frame 第一个数据到来以后使用的 DocumentLoader，这个时候，前一个主 Frame 的 DocumentLoader 已经不能再用(user agent 开始白屏，刷掉前一个页面的显示)。

5) 包含一个 HistoryController 对象，用于操作历史记录相关的接口，保存或者恢复 Document 和 View 相关的状态，维护前进后退队列，以实现前进后退功能，前进后退本质上是同 Page 对象关联的，FrameLoader 通过 HistoryController 操作 m_backFowardController 对象

6) 包含一个 ResourceLoadNotifier 对象，主要用于同 ResourceLoader 及 FrameLoaderClient 打交道，可以理解为 ResourceLoader 有事件变化或者发生的时候，通知 FrameLoader 的一个手段

7) 包含一个 SubframeLoader 对象，当 FrameLoader 下载的 Document 有子帧需要请求的时候(比如 HTMLDocument 中解析到 iframe 元素)，用来处理子帧请求

8) 将 FrameLoader 的状态封装到 FrameLoaderStateMachine 中，这个状态同 FrameState 不同，FrameState 倾向于判断 Frame 涉及的 Document 的下载状态，是出于发起状态 (Provisional)，还是出于已经收到响应但不全 (CommittedPage)，还是响应收全的状态，倾向于同 http 相关。而 FramLoaderStateMachine 倾向于同 DocumentLoader

相关，用来描述 FrameLoader 处理 DocumentLoader 的节点，是处于已经创建，还是显示的状态。

9) PolicyChecker 主要用来对 FrameLoader 进行一些校验。包括三种校验：NewWindow,Navigation 和 Content。NewWindow 对应于浏览器需要新开一个 tab 页或窗口的时候，Navigation 对应于一个页面请求发起的时候，Content 校验对应于收到数据以后(判断 Mime type 等)，PolicyChecker 通过提供对应的接口，由 FrameLoaderClient 来对这些请求进行校验，以确定是否允许继续，或者需要其它的动作。

3. 主要接口

Frame::init

功能：FrameLoader 的初始化

函数调用系列

→QWebFrame::QWebFrame(QWebPage* parent,QWebFrameData*frameData)

→QWebFramePrivate::init(QWebFrame* qwebframe,QWebFrameData*frameData)

→Frame::init()

→FrameLoader::init()

说明：主要做一些自身的初始化工作，比如初始化状态机，Sandbox Flags，创建 DocumentLoader 被设置为 Policy DocumentLoader 和

Provisional DocumentLoader ,调用 DocumentLoader 和 documentWriter 等的接口进行初始化操作

FrameLoader::commitProvisionalLoad

功能：提交 Provisional 阶段下载的数据

函数调用系列：

→DocumentLoader::finishLoading
→DocumentLoader::commitIfReady
→FrameLoader::commitProvisionalLoad

或者

→ResourceLoader::didReceiveData
→MainResourceLoader::addData
→DocumentLoader::receiveData
→DocumentLoader::commitLoad
→DocumentLoader::commitIfReady
→DocumentLoader::commitProvisionalLoad

说明：这个接口主要的操作是将 Provisional DocumentLoader 设置成 DocumentLoader，因为已经收到数据，所以 FrameState 也会跃迁到 FrameStateCommittedPage。还有历史记录，PageCache 相关的操作。

另外，这个接口会间接调用 FrameLoaderClientQt::transitionToCommittedForNewPage，通过

Frame::createView 创建出 FrameView 来。

Frame::finishedLoading

功能：frame 请求网络加载完成的时候调用此接口

函数调用系列

→ResourceLoader::didFinishLoading

→MainResourceLoader::didFinishLoading

→FrameLoader::finishedLoading

→FrameLoader::init()

说明：检查是否有网络错误，告诉 DocumentLoader 和 DocumentWriter 下载完成，以便进行后续操作（提交数据，解析）。

FrameLoader::finishedParsing

功能：解析完成调用此接口

函数调用系列

→DocumentWriter::end

→....

→Document::finishParsing

→....

→Document::finishedParsing

→FrameLoader::finishedParsing

FrameLoader::load(const ResourceRequest& request, bool lockHistory)

功能：加载一个 frame 请求，Frame 请求相关的数据，封装成 ResourceRequest 传入。

函数调用系列：一般由应用触发调用

说明：这个接口调用 FrameLoaderClientQt::createDocumentLoader 创建出 DocumentLoader，并以此 DocumentLoader 为 Policy Document Loader，进入 Policy check 流程。